**Coding Style Guide**

**Background**

C++ will be the programming language used to build the annotations tool program. `C++ is very good for making complex computer applications and graphical interfaces. Although this feature is great, this makes coding more difficult and bug prone compared to other languages like Java.

The purpose of this document is to provide a standard way of programming the annotation application to maintain simplicity and consistency in the code.

**C++ Version**

Code should be written in c++17 but optionally implementation of c++20 is readily available.

**Header Files**

Header files should be self-contained and end in .h

**Scoping**

**Local Variables**

Define local variables as close to first use as possible. This will include initializing variables when its declared.

**Member Functions**

Access modifiers should be used appropriately to limit the scope of functions. Add these accordingly so functions are used only where needed.

**Classes**

**Functions**

Functions that are not declared ‘void’ should have a return value

**Naming**

**File Names**

File names should be written with should begin with capital letters and use underscores to differentiate words. Examples include;

* My\_Annottations.cc

**Variable Names**

Hungarian naming format should be used for all variables. EG

* int getAge
* string fileName

**Function Names**

Should start with a capital letter and have a capital letter for each word. Eg;

* OpenFileDirectory
* WriteFile

**Class names**

Should start with a capital letter and have a capital letter for each word.

**Formatting**

**Tabs**

The tab function will be used to indent functions and nested functions. This will increase the readability of the code.

**Comments**

**Comment Style**

Use // for single line comments and /\* \*/ for multiple line comments

**Function Comments**

Before each new function, describe what it does using comments. For non-obvious, complex lines of code comments should be written before them.